

REMARKS

[0002] Applicant respectfully requests reconsideration and allowance of all of the claims of the application. The status of the claims is as follows:

- Claims 1-53 are currently pending
- Claims 1-13, 15-53 are amended herein

[0003] Support for the amendments to claims 1, 22, 27, 33, 38, 42, 47, and 50 is found in the specification at least at pages 8-9.

Cited Documents

[0004] The following documents have been applied to reject one or more claims of the Application:

- Wenocur: Wenocur et al, U.S. Patent Application Publication No. 2003/0009694
- Prentice: Prentice, Edward J., U.S. Patent Application Publication No. 2006/0268836
- Lennon: Lennon, Alison J., U.S. Patent Application Publication No. 2002/0152267

§ 103 Rejection of Claims 1-26

[0005] Claims 1-26 stand rejected under 35 U.S.C. § 103(a) as allegedly being obvious over Wenocur in view of Prentice. Applicant respectfully traverses the rejection.

Independent Claim 1

[0006] Applicant submits that the cited documents do not teach or suggest every element of claim 1. Specifically, neither Wenocur nor Prentice, alone or in combination, teaches or suggests the following, as recited in claim 1 (with emphasis added):

converting a plurality of programs into corresponding *intermediate language programs*, the plurality of programs being written in different programming languages but according to a common language specification.

[0007] Instead, Wenocur discloses a “storymail” system for developing electronic communication messages, or “stories” that may be emailed to a client device. (Wenocur, Abstract, Paragraph 67-71). However, since Wenocur’s disclosure is related to the development of electronic communication *messages*, the disclosure of Wenocur is silent with respect to the conversion programs of different languages to “*intermediate language programs*”, as recited in claim 1. (Emphasis added).

[0008] Second, Wenocur does not teach or suggest, as further recited in claim 1 (with emphasis added):

exposing resources of an operating system or an object model service *via functions of a program interface layer* to the plurality of intermediate language programs...wherein the program interface layer resides on top of a *common language run time layer that hands calls to the functions of the program interface layer* by the plurality of intermediate language programs to the operating system or the object model service for execution.

[0009] Instead, Wenocur discloses that the “applications 168” of its “storymail” system may include a “composition engine 170, a storyteller 172, email engine 173, and other applications 174” that may work cooperatively to build “story 180.” Nevertheless, even assuming, *in arugendo*, that each of the “applications 168” includes its own application program interface (API), Wenocur nevertheless does not teach or suggest that such APIs would be part of a single “program interface layer” that “*resides on top of a common language run time layer*,” as recited in claim 1. Indeed, even if each of the “applications 168” of Wenocur does in fact have its own API, such APIs would reside with the corresponding “applications 168” rather than on a *single* “program interface layer.”

[0010] Moreover, the conclusion that any APIs possessed by the “applications 168” of Wenocur are likely to reside at separate locations, rather than on a single “program interface layer”, is further supported by the fact that Wenocur discloses that the communication messages, or “stories” are intended to be sent from “story servers 302 and 328” to be consumed by “story enabled client 336”. (Wenocur, Paragraphs 122-223 and 166).

[0011] Accordingly, even if Wencour discloses APIs for various applications, they are likely to reside at *separate* locations (e.g., servers, clients), rather than on a “program interface layer” that resides on top of a “common language run time layer,” as recited in claim 1.

[0012] Moreover, the deficiencies of Wenocur with respect to these elements are not remedied by Prentice. Prentice discloses that codecs may be used to convert data streams, such as audio streams. (Prentice, Page 2, Paragraph 40). However, this

disclosure of Prentice does not teach or suggest the elements of claim 1 that are not taught or suggested by Wenocur.

[0013] Thus, for at least the foregoing reasons, the combination of Wenocur and Prentice does not teach or suggest every element of claim 1. Accordingly, claim 1 is believed to be allowable over the cited documents.

Dependent Claims 2-21

[0014] Claims 2-21 ultimately depend from independent claim 1. As discussed above, claim 1 is allowable over the cited documents. Therefore, claims 2-21 are also allowable over the cited documents of record for at least their dependency from an allowable base claim. These claims may also be allowable for the additional features that each recites.

Independent Claim 22

[0015] Applicant submits that the cited documents do not teach or suggest every element of claim 22. Specifically, Wenocur does not teach or suggest, as recited in claim 22 (with emphasis added):

a common language runtime layer running on top of the operating system that enables at least one intermediate language application to access at least one resource of the operating system.

[0016] Instead, Wenocur discloses a "storymail" system for developing electronic communication messages, or "stories" that may be emailed to a client device. (Wenocur, Abstract, Paragraph 67-71). However, since Wenocur's disclosure is related to the development of electronic communication *messages*, the disclosure of Wenocur

is silent with respect to a “*common language layer*” that runs on top of an operating system or “*intermediate language programs*”, as recited in claim 22. (Emphasis added).

[0017] Second, Wenocur does not teach or suggest, as further recited in claim 22 (with emphasis added):

an application programming interface layer running on top of the common language runtime layer, *the application program interface layer including functions* that enable the at least one *intermediate language application* to access the at least one resources of the operating system *via the common language runtime layer...*

[0018] Instead, Wenocur discloses that the “applications 168” of its “storymail” system may include a “composition engine 170, a storyteller 172, email engine 173, and other applications 174” that may work cooperatively to build “story 180.” Nevertheless, even assuming, *in arugendo*, that each of the “applications 168” includes its own application program interface (API), Wenocur nevertheless does not teach or suggest that such APIs would be part of a single “program interface layer” that “*resides on top of a “common language run time layer,”*” as recited in claim 22. Indeed, even if each of the “applications 168” of Wenocur does in fact have its own API, such APIs would reside with the corresponding “applications 168” rather than on a *single* “program interface layer.”

[0019] Moreover, the conclusion that any APIs possessed by the “applications 168” of Wenocur are likely to reside at separate locations, rather than on a single “program interface layer”, is further supported by the fact that Wenocur discloses that the communication messages, or “stories” are intended to be sent from “story servers 302

and 328” to be consumed by “story enabled client 336”. (Wenocur, Paragraphs 122-223 and 166).

[0020] Accordingly, even if Wencour discloses APIs for various applications, they are likely to reside at *separate* locations (e.g., servers, clients), rather included a “program interface layer” that resides on top of a “common language run time layer,” as recited in claim 22, that “enable the at least one *intermediate language application* to access the at least one resources of the operating system *via the common language runtime layer*. (Emphasis added).

[0021] Third, Wenocur does not teach or suggest, as further recited in claim 22 (with emphasis added):

a common language specification that enables a plurality of applications to be translated into intermediate language applications, the plurality of application including at least two applications that are written in different programming languages.

[0022] Once again, Wenocur discloses a “storymail” system for developing electronic communication messages, or “stories” that may be emailed to a client device. (Wenocur, Abstract, Paragraph 67-71). However, since Wenocur’s disclosure is related to the development of electronic communication *messages*, the disclosure of Wenocur is silent with respect to a “*common language specification that enables a plurality of applications to be translated into intermediate language applications*”, as recited in claim 22. (Emphasis added).

[0023] Moreover, the deficiencies of Wenocur with respect to these elements are not remedied by Prentice. Prentice discloses that codecs may be used to convert data

streams, such as audio streams. (Prentice, Page 2, Paragraph 40). However, this disclosure of Prentice does not teach or suggest the elements of claim 22 that are not taught or suggested by Wenocur.

[0024] Thus, for at least the foregoing reasons, the combination of Wenocur and Prentice does not teach or suggest every element of claim 22. Accordingly, claim 22 is believed to be allowable over the cited documents.

Dependent Claims 23-26

[0025] Claims 23-26 ultimately depend from independent claim 22. As discussed above, claim 22 is allowable over the cited documents. Therefore, claims 23-26 are also allowable over the cited documents of record for at least their dependency from an allowable base claim. These claims may also be allowable for the additional features that each recites.

§ 103 Rejection of Claims 27-53

[0026] Claims 27-53 stand rejected under 35 U.S.C. § 103(a) as allegedly being obvious over Lennon in view of Prentice. Applicant respectfully traverses the rejection.

Independent Claim 27

[0027] Applicant submits that the cited documents do not teach or suggest every element of claim 27. Specifically, Lennon does not teach or suggest, as recited in claim 27 (with emphasis added):

creating a plurality of groups for the set of API functions according to type, each group containing logically related API functions of the application program interface.

[0028] Instead, Lennon discloses the grouping of “multimedia items”, such as video, audio, and images, using XML schemas. (Lennon, Paragraph 71-80 and 212). This is apparent in the following section of Lennon, which states (with emphasis added):

The preferred arrangement assumes that all descriptions of *multimedia items* conform to a schema, and that schemas are expressed or represented using the W3C schema language, XML Schema. Individual descriptions are represented using XML document instances. XML Schemas are also represented as XML documents. Therefore descriptions (e.g. of *multimedia items*) can be stored along with their respective schemas in XML repositories or object stores. Alternatively, the descriptions can be stored in a database and effectively translated into XML documents when required.

(Lennon, Paragraph 71).

[0029] However, the grouping of “multimedia items” (e.g., video, audio, images) is not equivalent to the group of “API functions”, as recited in claim 27. Thus, Lennon does not teach or suggest “*creating a plurality of groups for the set of API functions according to type,*” as recited in claim 27.

[0030] Moreover, the deficiencies of Lennon with respect to this element are not remedied by Prentice. Prentice discloses that codecs may be used to convert data streams, such as audio streams. (Prentice, Page 2, Paragraph 40). However, this disclosure of Prentice does not teach or suggest the grouping of “API functions”, as recited in claim 27.

[0031] Additionally, since the combination of Lennon and Prentice does not teach or suggest, “*creating a plurality of groups for the set of API functions according to type*,” Lennon and Prentice also cannot teach or suggest, as further recited claim 27 (with emphasis added):

assigning a name to each group using the program development computer system, wherein *one of the groups includes at least one API function related to core concepts of the file system*, wherein *another of the groups includes at least one API function related to entities that a human being can contact*, wherein *another of the groups includes at least one API function related to document types that can be stored in the file system*, and wherein another of the groups includes at least one API function related to multiple kinds of media.

[0032] Thus, for at least the foregoing reasons, the combination of Lennon and Prentice does not teach or suggest every element of claim 27. Accordingly, claim 27 is believed to be allowable over the cited documents.

Dependent Claims 28-32

[0033] Claims 28-32 ultimately depend from independent claim 27. As discussed above, claim 27 is allowable over the cited documents. Therefore, claims 28-32 are also allowable over the cited documents of record for at least their dependency from an allowable base claim. These claims may also be allowable for the additional features that each recites.

Independent Claim 33

[0034] Applicant submits that the cited documents do not teach or suggest every element of claim 33. Specifically, Lennon does not teach or suggest, as recited in claim 33 (with emphasis added):

creating a first namespace with *application program interface (API) functions of an API layer* that enable identification of particular physical locations using the program development computer system, *the API interface layer running on top of a common language runtime layer to receive API function calls from an intermediate language program.*

[0035] Instead, Lennon discloses the grouping of “multimedia items”, such as video, audio, and images, using XML schemas. (Lennon, Paragraph 71-80 and 212). This is apparent in the following section of Lennon, which states (with emphasis added):

The preferred arrangement assumes that all descriptions of *multimedia items* conform to a schema, and that schemas are expressed or represented using the W3C schema language, XML Schema. Individual descriptions are represented using XML document instances. XML Schemas are also represented as XML documents. Therefore descriptions (e.g. of *multimedia items*) can be stored along with their respective schemas in XML repositories or object stores. Alternatively, the descriptions can be stored in a database and effectively translated into XML documents when required.

(Lennon, Paragraph 71).

[0036] Thus, Lennon, at best, describes creating named XML elements for “*multimedia items*” (e.g., video, audio, images), which is not equivalent to creating that “namespace” that includes “*API functions*”, as recited in claim 33. (Emphasis added). Thus, Lennon does not teach or suggest “creating a first namespace that includes *application program interface (API) functions of an API layer ... the API interface layer running on top of a common language runtime layer to receive API function calls from an intermediate language program.*,” as recited in claim 33. (Emphasis added).

[0037] Moreover, the deficiencies of Lennon with respect to this element are not remedied by Prentice. Prentice discloses that codecs may be used to convert data streams, such as audio streams. (Prentice, Page 2, Paragraph 40). However, this disclosure of Prentice does not teach or suggest the creating of namespace for “API functions”, as recited in claim 33.

[0038] Additionally, since Lennon and Prentice does not teach or suggest the above recited element of claim 33, Lennon and Prentice also cannot teach or suggest, as further recited claim 33 (with emphasis added):

creating a second namespace *that includes API functions of the API layer that enable identification of entities that can be contacted by a human being* using the program development computer system, wherein the first namespace and the second namespace are included in the file system, the file system being included in a programming interface.

[0039] Thus, for at least the foregoing reasons, the combination of Lennon and Prentice does not teach or suggest every element of claim 33. Accordingly, claim 33 is believed to be allowable over the cited documents.

Dependent Claims 34-37

[0040] Claims 34-37 ultimately depend from independent claim 33. As discussed above, claim 33 is allowable over the cited documents. Therefore, claims 34-37 are also allowable over the cited documents of record for at least their dependency from an allowable base claim. These claims may also be allowable for the additional features that each recites.

Independent Claim 38

[0041] Applicant submits that the cited documents do not teach or suggest every element of claim 38. Specifically, Applicant incorporates the reasoning presented above in response to the rejection of claim 33 under 35 U.S.C. § 103(a) to the extent that claims 33 and 38 recite similar subject matter. Based on this incorporated reasoning, Applicant also respectfully submit that Lennon and Prentice do not each or suggest, as recited in 38 (with emphasis added):

create a first namespace that includes application program interface (API) functions of the API layer that enable identification of particular physical locations, the API interface layer running on top of a common language runtime layer to receive API function calls from an intermediate language program; and

create a second namespace-that includes API functions of the API layer that are expected to be used by the first namespace and a plurality of additional namespaces, wherein the first namespace, the second namespace, and the plurality of additional namespaces are defined to organize a file system.

[0042] Thus, for at least the foregoing reasons, the combination of Lennon and Prentice does not teach or suggest every element of claim 38. Accordingly, claim 38 is believed to be allowable over the cited documents.

Dependent Claims 39-41

[0043] Claims 39-41 ultimately depend from independent claim 38. As discussed above, claim 38 is allowable over the cited documents. Therefore, claims 39-41 are also allowable over the cited documents of record for at least their dependency from an allowable base claim. These claims may also be allowable for the additional features that each recites.

Independent Claim 42

[0044] Applicant submits that the cited documents do not teach or suggest every element of claim 42. Specifically, Lennon does not teach or suggest, as recited in claim 42 (with emphasis added):

calling one or more first application program interface (API) functions of an API layer *that is running on top of a common language runtime layer, the one or more first API functions enable documents to be described.*

[0045] Instead, Lennon discloses use of an access scheme for "multimedia items", such as video, audio, and images. (Lennon, Paragraph 71-80 and 212). This is apparent in the following section of Lennon, which states (with emphasis added):

The preferred arrangement assumes that all descriptions of *multimedia items* conform to a schema, and that schemas are expressed or represented using the W3C schema language, XML Schema. Individual descriptions are represented using XML document instances. XML

Schemas are also represented as XML documents. Therefore descriptions (e.g. of *multimedia items*) can be stored along with their respective schemas in XML repositories or object stores. Alternatively, the descriptions can be stored in a database and effectively translated into XML documents when required.

(Lennon, Paragraph 71).

[0046] However, the access of an XML schemas for "*multimedia items*" (e.g., video, audio, images) is not equivalent to the group of "*API functions*", as recited in claim 42. (Emphasis added). Thus, Lennon does not teach or suggest "calling one or more first application program interface (API) functions of an API layer *that is running on top of a common language runtime layer*," as recited in claim 42. (Emphasis added). Indeed, Lennon is silent with respect to the "common language runtime layer" recite in claim 42.

[0047] Moreover, the deficiencies of Lennon with respect to this element are not remedied by Prentice. Prentice discloses that codecs may be used to convert data streams, such as audio streams. (Prentice, Page 2, Paragraph 40). However, this disclosure of Prentice does not teach or the above recited element of claim 42.

[0048] Thus, for at least the foregoing reasons, the combination of Lennon and Prentice does not teach or suggest every element of claim 42. Accordingly, claim 42 is believed to be allowable over the cited documents.

Dependent Claims 43-46

[0049] Claims 43-46 ultimately depend from independent claim 42. As discussed above, claim 42 is allowable over the cited documents. Therefore, claims 43-46 are also allowable over the cited documents of record for at least their dependency from an

allowable base claim. These claims may also be allowable for the additional features that each recites.

Independent Claim 47

[0050] Applicant submits that the cited documents do not teach or suggest every element of claim 47. Specifically, Applicant incorporates the reasoning presented above in response to the rejection of claim 42 under 35 U.S.C. § 103(a) to the extent that claims 42 and 47 recite similar subject matter. Based on this incorporated reasoning, Applicant also respectfully submit that Lennon and Prentice do not each or suggest, as recited in 47 (with emphasis added):

receiving one or more calls to one or more *first application program interface (API) functions of the API layer that is running on top of a common language runtime layer*, the one or more first API functions enable identification of entities that can be contacted by a human being.

[0051] Thus, for at least the foregoing reasons, the combination of Lennon and Prentice does not teach or suggest every element of claim 47. Accordingly, claim 47 is believed to be allowable over the cited documents.

Dependent Claims 48-49

[0052] Claims 48-49 ultimately depend from independent claim 47. As discussed above, claim 47 is allowable over the cited documents. Therefore, claims 48-49 are also allowable over the cited documents of record for at least their dependency from an allowable base claim. These claims may also be allowable for the additional features that each recites.

Independent Claim 50

[0053] Applicant submits that the cited documents do not teach or suggest every element of claim 50. Specifically, Applicant incorporates the reasoning presented above in response to the rejection of claim 42 under 35 U.S.C. § 103(a) to the extent that claims 42 and 50 recite similar subject matter. Based on this incorporated reasoning, Applicant also respectfully submit that Lennon and Prentice do not each or suggest, as recited in 50 (with emphasis added):

receive one or more calls to one or more *first application program interface (API) functions of the API layer* that enable identification of entities that can be contacted by a human being, *the API layer running on top of a common language runtime layer.*

[0054] Thus, for at least the foregoing reasons, the combination of Lennon and Prentice does not teach or suggest every element of claim 50. Accordingly, claim 50 is believed to be allowable over the cited documents.

Dependent Claims 51-53

[0055] Claims 51-53 ultimately depend from independent claim 50. As discussed above, claim 50 is allowable over the cited documents. Therefore, claims 51-53 are also allowable over the cited documents of record for at least their dependency from an allowable base claim. These claims may also be allowable for the additional features that each recites.

Closing Statement

[0056] In closing, Applicant's decision not to discuss the differences between the cited art and each dependent claim should not be considered as an admission that Applicant

concur with the conclusions set forth in the Office Action that these dependent claims are not patentable over the disclosure in the cited documents. Similarly, Applicant's decision not to discuss differences between the prior art and every claim element, or every comment set forth in the Office Action, should not be considered as an admission that Applicant concurs with the interpretation and assertions presented in the Office Action regarding those claims. Indeed, Applicant believes that all of the dependent claims patentably distinguish over the references cited. Moreover, a specific traverse of the rejection of each dependent claim is not required, since dependent claims are patentable for at least the same reasons as the independent claims from which the dependent claims ultimately depend.

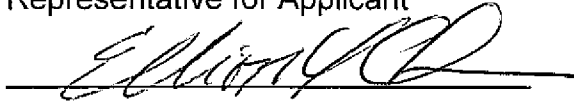
[0057] Furthermore, due to the Applicant's earnest belief that the claims, as rejected under Section 103(a), are allowable because their recited elements are not taught or suggested in the cited documents, Applicant will not address motivation to combine with respect to the claims during this response. However, Applicant hereby reserves the right to further challenge motivation to combine the cited documents.

Conclusion

[0058] For at least the foregoing reasons, all pending claims are in condition for allowance. Applicant respectfully requests reconsideration and prompt issuance of the application. If any issues remain that would prevent allowance of this application, Applicant requests that the Examiner contact the undersigned representative before issuing a subsequent Action.

Respectfully Submitted,

Lee & Hayes, PLLC
Representative for Applicant



Elliott Y. Chen
(elliott@leehayes.com; 206-876-6001)
Registration No. 58293

Dated: 1-23-10

Reviewer/Supervisor: Lewis C. Lee
(Lewis@leehayes.com ; 509-944-4711)
Registration No. 34656